

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-106366

(43)Date of publication of application : 22.04.1997

(51)Int.Cl.

G06F 12/00

(21)Application number : 08-217475

(71)Applicant : SUN MICROSYST INC

(22)Date of filing : 19.08.1996

(72)Inventor : SENATOR STEVEN T

(30)Priority

Priority number : 95 516429

Priority date : 17.08.1995

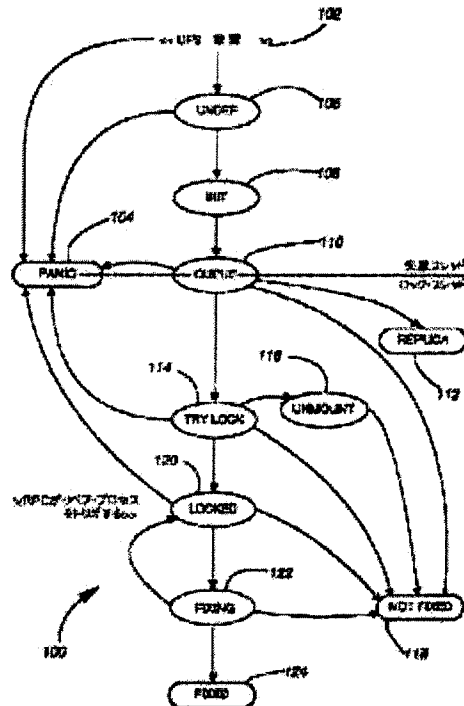
Priority country : US

(54) SYSTEM AND METHOD FOR RECOVERING FROM FILE SYSTEM PANIC FOR COMPUTER OPERATING SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a system/method for recovering from the file system panic of a UNIX file system and to increase the availability of the whole system.

SOLUTION: The system/method for recovering from the file system panic in a computer operating system, which is made for strengthening the UNIX file system 'UFS' and which increases the availability of the whole system by detecting a file system error, deciding whether on-line repair can be executed or not, locking/restoring the file system which breaks down and unlocking it are provided. Since average time until the fault of an independent thread increases by the reduction of time until the next fault and the average restoration time of a slave thread in time required for repair, the availability of the whole computer system increases. The opened system/method permit repair during use, calling is possible from a user level and the blocking of only the specified thread is permitted.



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-106366

(43) 公開日 平成9年(1997)4月22日

(51) Int.Cl.⁶

G 0 6 F 12/00

識別記号

5 3 1

庁内整理番号

F I

G 0 6 F 12/00

技術表示箇所

5 3 1 Z

5 3 1 R

審査請求 未請求 請求項の数18 O L (全 12 頁)

(21) 出願番号 特願平8-217475

(22) 出願日 平成8年(1996)8月19日

(31) 優先権主張番号 5 1 6 4 2 9

(32) 優先日 1995年8月17日

(33) 優先権主張国 米国 (U S)

(71) 出願人 591064003

サン・マイクロシステムズ・インコーポ
レーテッド

SUN MICROSYSTEMS, IN
CORPORATED

アメリカ合衆国 94043 カリフォルニア
州・マウンテンビュー・ガルシア アヴェ
ニュー・2550

(72) 発明者 スティーヴン・ティー・セナター

アメリカ合衆国コロラド州80920, コロラ
ド・スプリングス, ウェストミンスター・
ドライブ 8625

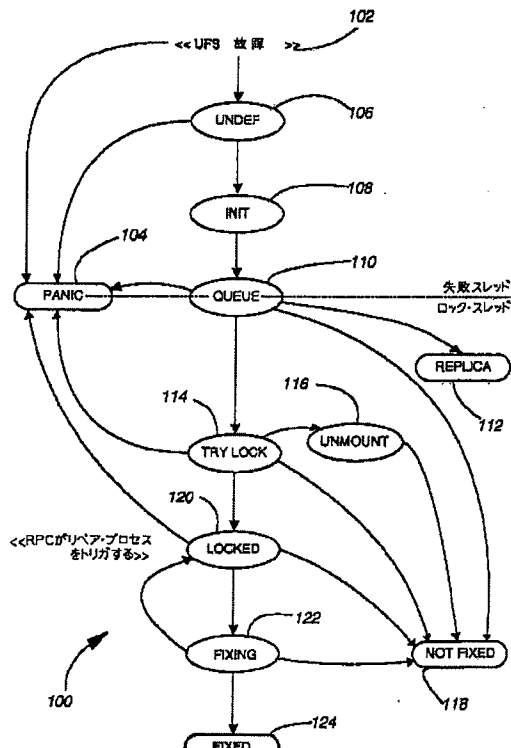
(74) 代理人 弁理士 湯浅 恭三 (外5名)

(54) 【発明の名称】 コンピューター・オペレーティングシステムのためのファイルシステム・パニック回復のシステムと方法

(57) 【要約】

【目的】 ユニックス・ファイルシステムのファイルシステム・パニック回復のためのシステム及び方法を実現して全システムの可用性を増加する。

【構成】 ユニックス・ファイルシステム「UFS」に対する強化から成り、ファイルシステム・エラーを検出して、オンライン修理ができるかどうか決め、それから故障したファイルシステムをロックし、修復して、アンロックすることによって、全システム可用性を増加する、コンピューター・オペレーティング・システムに対する、ファイルシステム・パニック回復のためのシステム及び方法である。次の故障までの時間と従属するスレッドの平均修復時間が修理に必要な時間だけに減少することによって、独立のスレッドの故障までの平均時間が増加するので、全計算機システムの可用性が増加する。開示されたシステムと方法は、使用中に修理することを許し、ユーザーレベルから呼ぶことができ、特定のスレッドだけのブロッキングを許す。



【特許請求の範囲】

【請求項1】 コンピューター・オペレーティング・システムの故障したファイルシステムを選択的に誤りロックする方法であって、以下のステップから成っている：前記コンピューター・オペレーティング・システムの故障したファイルシステムにおけるエラーを検出するステップ；前記故障したファイルシステムのオンライン修復ができるかどうか決めるステップ；前記オンライン修復ができるならば、故障したファイルシステムをロックするステップ；前記故障したファイルシステムの前記オンライン修復を成し遂げるステップ；前記故障したファイルシステムをアンロックするステップ。

【請求項2】 請求項1に記載の方法において、前記検出するステップが、次のステップからなる：前記故障したファイルシステムの矛盾に注目するステップ；前記故障したファイルシステム矛盾に対応する故障レコードをつくるステップ；オペレーティング・システム・ロック・スレッド上に前記故障レコードを置くステップ。

【請求項3】 請求項2に記載の方法において、前記矛盾が、オペレーティング・システム故障スレッドにตอบสนองして注目されることを特徴とする方法。

【請求項4】 請求項1に記載の方法において、前記ロックステップが、次のステップから成る：関連するコンピューターの大容量記憶装置の上で、前記故障したファイルシステムと関連するデータブロックをアクセス不可能と記録するステップ。

【請求項5】 請求項1に記載の方法において、前記オペレーティング・システムが、前記故障したファイルシステムのオンライン修復ができるかどうか決めるステップが不可能であることにตอบสนองしてシャットダウンされることを特徴とする方法。

【請求項6】 請求項1に記載の方法において、更に次のステップを含む：前記故障したファイルシステムがロックされる一方、他のオペレーティング・システム・ファイルシステムに対応している他のスレッドが継続するのを許すステップ。

【請求項7】 以下の要素からなるコンピューター・プログラム・プロダクト：コンピューター・オペレーティング・システムの故障したファイルシステムを選択的に誤りロックするためにコンピューター可読のコードを記録されたコンピューターが使用できる媒体；コンピューターに、前記コンピューター・オペレーティング・システムの故障したファイルシステムにおけるエラーの検出をさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムのオンライン修復ができるかどうかの決定をさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記オンライン修復が可能ならば前記故障したファイルシステムのロックをさせるように

するために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムのオンライン修復をさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムをアンロックさせるために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項8】 請求項7に記載のコンピューター・プログラム・プロダクトにおいて、前記コンピューターにエラーの検出をさせるようにするために構成された前記コンピューター可読のプログラムコード・デバイスが更に以下の要素からなる：コンピューターに、前記故障したファイルシステムにおける矛盾に注目することをさせるために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、前記故障したファイルシステムの矛盾に対応している故障レコードをつくることをさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス；コンピューターに、オペレーティング・システム・ロック・スレッド上に前記故障レコードを置くことをさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項9】 請求項7に記載のコンピューター・プログラム・プロダクトにおいて、前記コンピューターにロックをさせるようにするために構成されたコンピューター可読のプログラムコード・デバイスが更に以下の要素からなる：コンピューターに、関連するコンピューター大容量記憶装置の上で、前記故障したファイルシステムと関連するデータブロックをアクセス不可能であると記録させるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項10】 請求項7に記載のコンピューター・プログラム・プロダクトにおいて、更に次の要素を含む：コンピューターに、前記故障したファイルシステムのオンライン修復ができないという決定に答えて前記オペレーティング・システムをシャットダウンさせるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項11】 請求項7に記載のコンピューター・プログラム・プロダクトにおいて、更に次の要素を含む：コンピューターに、前記故障したファイルシステムをロックする一方、他のオペレーティング・システム・ファイルシステムに対応している他のスレッドの続行を許させるようにするために構成されたコンピューター可読のプログラムコード・デバイス。

【請求項12】 アプリケーションプログラムを実行するために、その上にロード可能なコンピューター・オペレーティング・システムを含むコンピューターにおいて、前記オペレーティング・システムは、前記アプリケーションプログラムからアクセス可能な複数のファイル

システムを持っており、次の要素から成る：前記複数のファイルシステムの内の故障したものにおけるエラーを示すための故障スレッド；前記故障スレッドにตอบสนองして始まったロック・スレッド；もしそれについてオンライン修復ができるなら、前記故障したファイルシステムを選択的にロックするために前記ロック・スレッドにตอบสนองする誤りロック；オンライン修復が実行されているとき、前記誤りロックにตอบสนองして前記複数のファイルシステムの内の前記故障したものに他のスレッドがアクセスするを禁止し、これによって前記故障したファイルシステムのオンライン修復の実施にตอบสนองして削除されるブロック。

【請求項13】 請求項12に記載のコンピュータにおいて、前記オペレーティング・システムが次の要素から成る：前記アプリケーションプログラムに前記オペレーティング・システムをインターフェースするためのユーザ層；前記ユーザ層の基礎をなしているカーネル層；前記カーネル層の基礎をなしていて、それらの間でファイルシステム・インタフェースを定義しているファイルシステム層。

【請求項14】 請求項12に記載のコンピュータにおいて、前記ブロックが、前記ファイルシステム・インタフェースで実現されることを特徴とするコンピュータ。

【請求項15】 請求項13に記載のコンピュータにおいて、前記ファイルシステム層が、ユニックス・ファイルシステム層から成ることを特徴とするコンピュータ。

【請求項16】 請求項13に記載のコンピュータにおいて、前記ファイルシステム・インタフェースが、VF 30 S層から成ることを特徴とするコンピュータ。

【請求項17】 請求項12に記載のコンピュータにおいて、更に次の要素を含む：前記複数のファイルシステムと関連するデータのブロックを記憶するための、前記コンピュータと関連するコンピュータ大容量記憶装置。

【請求項18】 請求項17に記載のコンピュータにおいて、前記故障したファイルシステムと関連する前記コンピュータ大容量記憶装置上の選ばれたデータのブロックが、前記ロック・スレッドにตอบสนองしてアクセス不 40 可能であると記録されることを特徴とするコンピュータ。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 一般に本発明は、コンピュータ・オペレーティングシステム「OS」のファイルシステム「FS」の分野に関連がある。特に、本発明は、ファイルシステム「パニック回復」に対する「誤りロック」を選択的にインプリメントし、削除する方法およびシステム 50 に関連する。これは、ユニックス・システムVファイル

システム（「VFS」、同様に「VFSop」又は「Vnode」として名を知られている）階層又は同様のOSファイルシステム・インターフェースを含んでいるコンピュータ・オペレーティングシステムに関する特別なユーティリティである。

【0002】

【従来の技術】

【0003】

【発明が解決しようとする課題】 コンピューターの大量 10 記憶媒体上の名をつけられたオブジェクトを記憶したものであるようなファイルのグループ化として、ファイルシステムは定義できる。このように、ファイル・システムは、そのファイルシステムが適当に作用するために、お互いが全く一致していなければならない一組のメモリー上とディスク上のデータ構造を保守する。これらのデータ構造が一貫していないとき、例えばユニックス・オペレーティングシステムは、強制的なトータルシステム 20 しゃ断「パニック」を実施する。それから、この「パニック」は、コンピュータ・システムをシャットダウンし、全てのディスク上の矛盾の修復と正常な系統運用のリジュームをするために必要な時間の間、全部のシステムを利用できなくする。

【0004】 ユニックスへのロックFS「lockfs」機能性の追加を通して、任意のファイルシステムのために一般的にインプリメントされてもよい全ファイルシステムをロックする能力が、提示された。この特長は、ロックの異なる種類に従って選ばれたオペレーションを抑止する能力を提供する。例えば、ファイルシステムは、抑止されていた他のいかなるオペレーションもなく、一時的 30 かつ選択的に変更不可能にされることができ（例えば、ここでファイルが、削除されるのを妨げられることができる）。「抑止された」オペレーションは、ファイルシステムが再びロックを解かれるまで単に待つだけのものである。この点について、「削除ロック」又は「ネーム・ロック」は、ファイルシステムが、依然として、ほんの程度の定義済み方法だけにおいて変わることができ 40 る一種のロックである。ロックされたファイルシステムへのアクセスは、そのファイルシステムのロックが解除されるまでの認め得るディレイを生ずる。

【0005】 これに対して、「ハード・ロック」は特定のファイルシステムが単に更なるアクセスから除外されるものである。例えば、記憶媒体欠陥が検出され、データ・リカバリ動作が実現されなければならないとき、ハード・ロックは起動される。ハード・ロックは、ネームスペースからファイルシステムを削除しない、しかし、エラーにより失敗したその内部の何れへもアクセスし、そしてハード・ロックをクリアする方法は、ファイルシステムをアンマウントすることだけである。特に、与えられたコンピュータ上のファイルシステムが整合性問 50 題を持っていれば、コンピュータは「ダウン」し、そ

れが提供するサービスの全てが、全てのユーザーから利用できなくなる。この整合性問題に遭遇するのが、偶然ネットワーク上のサーバ・コンピュータであるとき、これは、既存のクライアント-サーバ計算機システムにおいて特に緊急で高価な状況である。

【0006】システム故障時間を避ける試みにおいて、ファイルシステムそれ自身にチェック・アルゴリズムを埋めこむことは、以前に提案された。この手法の典型は、チェックが二相コミット・プロトコルを利用しているコアにおいてされる、ヴェリタス (Veritas) からのVXFSソフトウェアと、よりデータベース風のモデルに構築したIBMジャーナルド・ファイルシステム「JFS」である。いずれにせよ、チェック機能をインプリメントするためにファイルシステムに追加した、追加の中央処理装置 (CPU) 「オーバーヘッド」が、全体的により遅いシステム動作をもたらした。アウスベックス (Auspex) によって実現されたもう一つのチェックのアプローチは、単にファイルシステム・オペレーションを提供するだけに特化されたCPU上で動作するアルゴリズムを利用する。従って結果として生じるシステムは、それが設計された特別なハードウェア・インプリメンテーションに特有であり、したがって他のアーキテクチャに対して限られた適用可能性しか備えていない。

【0007】

【課題を解決するための手段】本発明のシステムと方法は、ファイルシステム「パニック回復」を実現するために用いられる「誤りロック」と命名される代替ファイルシステム・ロッキング機構を既存の「書き込み」及び「ハード」ロックに提供する。本明細書において開示されたように、誤りロックは (ハード・ロックのように) すべてをロックするが、ファイルシステムが一貫させられたあと、(書き込みロックのように) 依然としてロックを解除されることができる。本発明のシステムと方法を利用すれば、エラーに遭遇した特定のファイルシステムと、そのファイルシステム・サービスを用いていた特定のユーザーが影響されるだけで、これにより独立したユーザーにとっての全システムの可用性が大いに増加する。

【0008】本発明のファイルシステム・パニック回復システムと方法は、ユニックス・ファイルシステム「UFS」を拡張して、ファイルシステム・エラーを検出して、オンライン修復ができるかどうか決め、それから故障したファイルシステムをロックして、修復し、そしてアンロックすることによって全計算機システム可用性を増やすために役に立つ、特別なユーティリティである。本明細書において明らかにしたシステムと方法は、VFS層またはその同等物、任意のシステムVベースのユニックス・システム、IBMのAIXまたはマイクロソフトのNTオペレーティングシステムを含んでいるオペレーティングシステムの上で、同様に有利に実現することができる。

【0009】本発明のシステムと方法が、工夫されたあるオンライン・ファイルシステム・チェックへのアプローチを提供する既存のロック機構の付属物は、モノリシックの上でカーネルを実現して、マルチプロセッサ上の実行であることができる。それは、使用中にファイルシステム修繕でき、ユーザーレベルから呼ぶことができ、そのうえUFSロックファイル・システム「lockfs」層に、一般のファイルシステム・データ構造とアルゴリズムに関して非侵入的であると同時に特定のスレッドだけを抑止させるように修正することができる。

【0010】本明細書において明らかにしたようなコンピュータ・オペレーティングシステムの故障したファイルシステムを選択的にエラーロックするための特定の方法的インプリメンテーションとその方法を実現するコンピュータ・プログラム・プロダクトにおいて、この方法は、コンピュータ・オペレーティングシステムの故障したファイルシステムにおけるエラー検出し、故障したファイルシステムのオンライン修復ができるかどうか決めるステップから成る。この方法は更に、オンラインで修復できるなら故障したファイルシステムをロックし、故障したファイルシステムのオンライン修復を成し遂げて、故障したファイルシステムをアンロックするステップから成る。より特定のインプリメンテーションにおいて、この検出するステップは、故障したファイルシステムにおける矛盾に注目し、故障したファイルシステムの矛盾に対応する故障レコードをつくって、その故障レコードをオペレーティングシステムのロック・スレッドに置くステップによって遂行される。

【0011】同様に、アプリケーションプログラムからアクセス可能な複数のファイルシステムを持っているオペレーティングシステムにより、アプリケーションプログラムを走らせるのために、その上にロード可能なコンピュータ・オペレーティングシステムを含むコンピュータを開示する。オペレーティングシステムは、複数のファイルシステムの内の故障したものにおけるエラーを示す故障スレッドと、故障スレッドに応答して開始されたロック・スレッドとを含む。それについて、オンライン修復ができるならば、誤りロックは故障したファイルシステムを選択的にロックするためのロック・スレッドに応答する。複数のファイルシステムの内の故障したもののオンライン修復が実行されている間、それに他のスレッドがアクセスすることを禁止するために、ブロックが誤りロックに応答し、その後、ブロックは故障したファイルシステムのオンライン修復の完成に応答して削除される。

【0012】本発明の前記及び他の特長と目的が、そして、それらを達成する方法がより明白になるだろう。そして発明それ自身が添付の図面に関連してとられた好適な実施例の以下の記述の引用によって、最もよく理解できるだろう。

【0013】

【実施例】本発明が用いられるシステム環境は、汎用コンピュータ、ワークステーション・パソコンが、いろいろな型の通信リンクを通して、クライアント-サーバ構成において接続される、一般の分散された計算機システムを包含し、ここで（多くはオブジェクトの形の）プログラムとデータが、システムのいろいろなメンバによって実行のために利用可能であり、システムの他のメンバによってアクセス出来るようにされる。汎用ワークステーション・コンピュータの要素のいくつかは、図1において示される。ここで、入出力(I/O)セクション2、中央処理装置(CPU)3、及びメモリ・セクション4を持っているプロセッサ1が示される。I/Oセクション2は、キーボード5、表示装置6、磁気ディスク装置9とコンパクトディスク・リードオンリーメモリ(CDROM)駆動装置7に接続している。CDROM装置7は、典型的にプログラム10とデータを含むCDROM媒体8を読みとることができる。本発明の装置と方法を成し遂げるために機構を含んでいるコンピュータ・プログラム製品は、そのようなシステムのCDROM 8上、磁気ディスク装置9上、またはメモリ・セクション4に置くことができる。

【0014】図2を参照すると、図1において記載された複数のコンピュータのネットワークから成っている計算機システムでのアプリケーション・プログラムの実行の基礎をなしている妥当なユニックス・オペレーティングシステム層20の簡略概念上の図解が示される。図示された詳細は、「ユーザ」(システム・コール)層22、「カーネル」24と「UFS」層26である。カーネル24内部には、UFSモジュール28と、いくつかの例としてネットワークファイルシステム「NFS」モジュール30が存在する。UFSモジュール28は、例えばファイル命名、記憶領域割付けと他のサービスを含むことができる。図示のように、(Op1、Op2およびOp3のような)たくさんのオペレーションがVFS層32でUFSモジュール28とNFSモジュール30に生じることができる。

(VFSopまたはVnode層としても知られている) VFS層32は、いろいろなオペレーションを実行することができる、カーネル24とUFS層26の間のインターフェースである。

【0015】図3を参照すると、以前に図2において記載したVFS層32を含む、(ユーザー層22、カーネル24とUFS層26から成る)システム層20が示される。それに加えて、記憶装置40が、たくさんのUFSスレッド38とともに示される。UFSスレッド38は、実行しているプログラムまたはプログラムコードを実行することができる独立の「もの」として都合よく考えることができる。(これは複数のスレッドを持つことができる独立のものである「プロセス」との対比である)。

【0016】記載した正常動作において 特定のファイ

ルシステム・オペレーションを要求するアプリケーションプログラムの実行に応じて、UFSスレッド38がユーザー層22、カーネル24とVFS層32を介してUFS層26に通る。さらに図4を参照すると、ファイルシステム故障の発生時の以前の図のシステム層20と記憶装置40が示される。この点について、以下の定義を参照されたい：「Authoritative System Reference (命令的システム参照)」または「ASR」はシステムの修正挙動を生じる仮想のエンティティである。

10 【0017】「Error (エラー)」はASRによって生じたシステム挙動と実際のそれとの間の差である。

【0018】「Fault (フォールト)」はエラーを起こす可能性を持つ原因である。「Failure (故障)」はフォールトの表明である。

20 【0019】「UFS Failure」はその明確に示された動作からそれているUFSのインスタンスである。あるインプリメンテーションにおいて、UFS故障は、「synchronous failures (同期故障)」(プログラムコードが、明示的にチェックする)か、「asynchronous failures (非同期故障)」(トラップ処理機構によって取り扱われる)のいずれかである。UFS故障の後の型によって提示された問題は、本明細書において開示されたシステムと方法に適用できない。

30 【0020】サン・マイクロシステムズ社(本発明の譲り受け人)によって開発され、ライセンスを与えられるソラリス・オペレーティングシステムの、特定のインプリメンテーション内では、同期故障の2つの型に遭遇することができる：「Assertions (アサーション)」または「Asserts (主張)」はプリ・プロダクション・ソフトウェアにおいてだけチェックされる条件式である。真であるとき、システムは強制的にシャットダウンされる。

【0021】「Panics (パニック)」は常に、チェックされる条件式である。それらは「永久的なアサーション」であり、語「panic (パニック)」は「強制されたシステム・シャットダウン」を意味するための名詞、そして「シャットダウンにシステムを強制する」ことを意味する動詞としてしばしば用いられる。

40 【0022】「Failing Thread (故障スレッド)」はUFS故障が現れる状況におけるスレッドである。従属する(そして、独立の)スレッドが、故障ファイルシステムのリソースに依存する(またはしない)スレッドとしてそれぞれ、定義される得る。

【0023】「Availability (可用性)」はシステムが受け入れられる応答時間以内にサービスを提供することができる予想された時間の部分である。重大な影響を伴わずに短期間サービスを与えず、または遅らせることができるシステムを記述することが出来る。それは量的に次のように定義できる：

$$Availability = \frac{MeanTimeToFail}{MeanTimeToFail + MeanTimeToRepair}$$

または代わりに：

$$Availability = \frac{1}{\left(1 + \frac{MeanTimeToRepair}{MeanTimeToFail}\right)}$$

本発明のシステムと方法は、システムのユーザーを故障*

*したファイルシステムに従属するユーザーと従属しないユーザーとに分割して、可用性の粒状性を全システムからファイル・システム・サービスのそれへ減らすことを仮定している。

【0024】表1：システム・コール故障セマンティクス：ローカルおよびリモート・スレッド

スレッド	システム・コール故障セマンティクス
ローカル+リモート：故障、従属	リターン・エラー (ERESTART)
ローカル：従属	アンロック又はアンマウントまでブロック
リモート：従属	リターン・エラー (EWOULDBLOCK)
ローカル+リモート：独立	故障なし：遅延なし

従来のUFSインプリメンテーションにおいて、多量の状態は、インコアで保持された。しかし、この状態の全てが、UFSそれ自身によって管理されるのではない。UFS故障は、このインコア状態の少なくともいくつか信頼できないことを示す。これから回復するために、以下の動作が、必要である：

(1) 故障されたファイルシステムは、静止されなければならない；

(2) 不良インコア状態は、破棄されなければならない；

(3) ディスク上の状態は、少くとも一貫しているように、そして（多分）訂正するために検査されなければならない；

(4) インコア状態は、一貫した値から再初期化されなければならない；そして、

(5) ファイルシステム静止で「止まっていた」オペレーションは、進むために解除されなければならない。

【0025】特にユニックス・オペレーティングシステムにおいて、この問題はpanic()機構によって言及される、そして、全てのインコア状態は破棄される。修復と再初期化は、再起動時に起こる。必要なデーモンが再開されるまで、プロトコルの性質によって、ネットワークファイルシステム・オペレーションは、停止している。破棄されたローカル操作は、消失する。fsckによってされる修復を除いてlockfsは上記の機能の全てを実行する。UFS故障は、UFS内部の多くの異なる論理レベルで遭遇されることができる。このように、どの複数または単数のリソースのロックが故障の時点で保持されているかに関する保証がない。これは、仮想計算機のようなUFSが依存する他のサブシステムと同様にUFSリソースとロックに対して真である。伝統的なpanic()ルーチンは、ロック獲得アルゴリズムにおける明示されたコードによって、この問題に対処する。パニックのときは常に、ロック要求は許可され、他の非パニック・スレッドは、実行するのを防げられる。

【0026】lockfsは、ioctl()システム・コールから起動される。したがって、それはより階層的にリソースとロックに関して作用する。しかし、特別なケースのコードをUFS lockfsとそれによって起動できる他の全てのルーチンに入れることは、追加の機構の採用を必要とし、一般的UFSデータ構造とアルゴリズムに強制する。

【0027】UFSインプリメンテーションは、アロケーション・ビットマップと要約構造内部の冗長情報を含む。これの効果は、故障が実際のエラーが生じた時より非常に後の時間に生じる傾向があるということである。したがって、エラーと故障条件「パニック・ストリング」の間に多対1関係がある。したがって、提案された機構は、検出された全ての悪い構造を破棄し、そしてそれらを一貫した値に再初期化する。結果として、常にそれは故障に対処する際の現在のパニック・再起動メカニズムと同様である。

【0028】故障の影響があまりに激しいとき、あるいは、エラーそれ自身が解決できるときは、修復を試みることは適当でない。これらの影響は、MTTFかMTTRが受け入れ不可能に高レベルであるように表せる。他の表現をすれば、故障が、あまりにしばしば生じているか、あるいは、故障があまりにたくさんの被害を引き起こす。これは、一組の新しい調整可能な変数によって制御可能である。エラーが再現可能であるならば、エラーをマスクするために本発明のファイルシステム・パニック回復を用いるよりは、むしろバグをフィックスする方が正しい。本明細書において開示されたシステムのユーティリティと方法は、間欠的に生じて、まだ再現可能でないエラーに対処するためのものである。

【0029】一般的なケースにおいて、UFSファイルシステムを訂正する試みが、なされるべきである。しかし、いくつかの特定のUFSファイルシステムは、それらに課された追加の制約を持つ。ルート「/」及びユーザ「/usr」ファイルシステムはfsckを起動する必要がある。結果として、それらをロックすることは、デッドロ

ック条件を引き起こす。活動中のカーネル・アカウントファイルが在り、あるいはスワッピングが生じることが出来るUFSファイルシステムも、同様のデッドロック条件をつくることのできる。したがって、これらの特定のファイルシステム上で、パニックは起動される。

【0030】世代番号が変化したinodesは、lockfs調和アルゴリズムによって認識される。古いinodeは、それを強制的にアンマウントしたファイルシステム上のファイルの参照に現れさせるために修正される。新しいinodeモードまたはアルゴリズムは、必要でない。割当てinode (ファイルシステムの割当てファイルと関連させられたinode) が削除されるならば、割当て施行はこのファイルシステム (closedq()に類似した方法において) の上で使用禁止にされる、そして、警告メッセージは記録される。これが「fsck-p」のためのケースで決してなく、手動の修復がinodeが削除されるようにするために必要であることに注目されたい。アンリンクされたが、依然として参照されているファイルは検出可能で、削除されないだろう。Fsckの出力は、ロガー・コマンドによって獲得されて、syslogファイルに送られる。

【0031】図4の典型的な例で示すように、ファイルシステム故障のイベントで、故障スレッド42は、ユーザ層22、カーネル24を介してUFS層26へ通る。故障スレッド42の検出時に、参照番号44で示されたように、UFS故障レコードがつくられ、ロック・スレッド46の作業待ち行列の上に置かれる。故障が生じた時ロック・スレッド46は始まり、そしてそれは、本明細書の下文に記述されるように、記憶装置40上のブロックにその後はアクセスできないと記録して、そのファイルシステムをロックするように作用する。

【0032】参照番号48で示されるように、それから、故障スレッド42は、エラーを持って復帰する。応答して、参照番号50において、記憶装置40で、故障したファイルシステムのスーパーブロックのfs_cleanフラグは不良と記録され、そして、参照番号52によって示されるように、誤りロック36が不良のファイルシステムの上に置かれる。示されるように、誤りロック36は、VFS層32で他のUFSスレッド54を抑止するのに役立つ。参照番号56で、lockfsが復帰する時、UFS故障は状態「LOCKED」へ遷移する。

【0033】ファイルシステムの矛盾が検出されると、以前に記述されたプロセスが起動される。しかし、矛盾はその瞬間に起こることができず、そして、いつそれが生じたかを示す十分な情報がその時点で利用可能でないで、アプリケーション・チェッカーを必要とする。lockfsに関して、与えられたファイルシステムのための、全てのカーネル24の状態 (記憶ブロックと関連する「管理作業」と例えばファイル名) は、スタップを除いて破棄される。「アンロック」プロセスは調和と呼ぶ。ロック・ディスク機能をトリガーし、そして、記憶

装置40から現在の状態を検索する。そして、実際それは、その後に残された以前に使用中だったものための全てのスタップのために記憶装置40から現在の状態を検索するだけである。したがって、ファイルが変化するならば、記憶装置40上のファイルの状態は、正確だと仮定される。ファイルがいくらか削除されたならば、エラーは後に検出される、しかし、それは単なるアプリケーション・エラーであって、ファイルは単に再び書込まれる。lockfsによって可能化された調和は、このチェックを許す。

【0034】この文脈において、図4は概念的に内部のデータ構造を図示する、しかし、問題があることを実際に識別するプロセスはエラーコード「erestart」を参照番号48で提供する。このエラーコードは、いくつかのアプリケーションのケースにおいて、チェックがその特定のエラーコードに対して行われ、同じオペレーションが再びユーザに透過的に試みられるだろうことを意味する。それにもかかわらず、全てのオペレーションが再実行できる訳ではない。例えば、ファイルをつくらうとする際に、いくつかのケースにおいては、ファイルをつくることは、ファイルシステムが損害を与えられた仕方によって非常に難しい。

【0035】さらに図5を参照すると、ロック・スレッド46と関連する図4のシステム層20が示され、修復プロセスを始めるために参照番号58によって示されるように、RPCがつくられる。参照番号60で、inetdはRPCの生成を通知され、参照番号62で示されるように、UFSデーモンが始まる。応答において、fsckプロセスが参照番号64で始まる。その間に、RPCが非同期だったので、参照番号66で示されるように、inetdが復帰する。参照番号68で、fsckが記憶装置40上のスーパーブロックに修復するように記録し、そして、参照番号70でUFS故障レコードが「修復中」状態へ遷移する。

【0036】以前に記述されたように、ロック・スレッド46は、修復プロセスを始める。それは、ファイルシステム一貫性チェッカーをカーネル24から呼んで実行させ、それから、それが実際に始めることを確実にするようチェックする。言い換えると、ファイルシステムがエラーでロックされているならば、lockfsのために以前に記述されたように、記憶装置40上に残されたある程度のキーがある。決してコンピューターをデッドロックさせないことは、ロック・スレッド46のジョブの一部である。従って、所定の期間内に何も起こらないならば、それがエラーを発生させ、システム管理者がそれを単にサービスの遅延と見えるよりは、注意すべきものと認識するだろうから、コンピューターはシャットダウンされる。一貫性チェッカーを呼ぶことによって、システムは前進し、ファイルシステムを修復する。

【0037】さらに図6を参照すると、参照番号72で fsckがファイルシステム修復を完了し、記憶装置4

0の上でファイルシステムに「クリーン」と記録する。それからFscckが、参照番号74で同様にlockfs ioctlに「アンロック」を発行する。応答において、参照番号76でlockfsサブモジュールが、UFS故障レコードに「修復済み」と記録するためにパニック回復ルーチンに呼ぶ。参照番号78でlockfsは戻り、他のスレッド80が、誤りロック36（図4）の除去によってUFS層26まで続く事を許される。

【0038】図7を参照すると、以前に図3-6に関して記述された本発明の方法とシステムの特定のインプリメンテーションに従うUFS故障のイベントにおける複数の状態の間の可能な遷移100を図示している状態図が示される。

【0039】イベント102で、UFS故障が、検出された。Struct ufs_failure（又はtypedef uf_t）レコードが、各UFS故障と関連させられる。ファイルシステムが故障した時、ufsvfspから現在のstruct ufs_failureへのポインタが、ある。これらは、struct ufs_queueによって記述され、UFSスレッドによって用いられる待ち行列に置かれる。このスレッドは、待ち行列上に非ターミナルのufs故障があるときにのみ存在するだけだろ

```

struct buf*uf_bp; /*スーパーブロックを含んでいるptrからbufへ*/
/*スーパーブロック状態をアップデートするために*/
/*用いられる*/

kmutex_t *uf_vfs_lockp; /*ufsvfspを検査し修正するために用いられる*/
struct vfs_ufs *uf_vfs_vufp; /*これが与えられたfsの複製、又は*/
/*オリジナルの故障であるかどうか決める*/
/*ために用いられる；同様にfsにつき*/
/*あまりに頻繁な故障を防ぐために*/
/*用いられる*/

struct vfs *uf_vfsp; /*lockfsを起動し、fsにネームを得ること*/
/*に必要とされる構造を得るために用いられる*/

struct ufsvfs *uf_ufsvfsp; /*non-terminal状態において*/
/*関連させられたufs_failureがある時に*/
/*lockfsを起動し、fsがunmounted/remounted*/

/*であるかどうかを決めるために*/
/*必要とされる構造を得るために用いられる*/

char uf_fsname[MAXMNTLEN]; /*スーパーブロックからコピーされる；*/
/*fsがunmountedならば、エラー*/
/*メッセージを発生させるために用いられる*/

char uf_panic_str[LOCKFS_MAXCOINENTLEN]; /*パニックに移るならarg値を*/
/*含むオリジナルのパニックメッセージ*/
/*を印刷するために用いられる*/

```

【0044】状態110「QUEUE」は、これらのレコードがロック・スレッドのための作業待ち行列の上へ置かれたことを意味する。ロック衝突のために、いくつかのケースにおいてレコードが作業待ち行列の上に置かれ

*う。UFSスレッドとその関連させられた待ち行列の間には1対1の対応がある。各故障は、関連する状態を持つ。

【0040】端末状態104、「PANIC」はファイルシステムが修復を許されなかった、ロックできなかったか、またはUFSパニック回復アルゴリズムが内部エラーに遭遇したことのいずれかを意味する。それは、全システムに対する端末状態である。

【0041】状態106「UNDEF」は、これらのレコードが新たに割り当てられて、まだ初期状態にされなかったことを意味する。それらの存在は、故障が生じたことを意味する。これらは、故障しているスレッドのコンテキストにおいて割り当てられる。

【0042】状態108「INIT」は、これらのレコードが初期状態にされたことを意味する。それらの存在は、このファイルシステムがロックされる事を許され、修復できることを意味している。これらは、故障しているスレッドのコンテキストにおいて初期状態にされる。以下のUFS制御構造とポインタが、struct ufs_failureにコピーされる。

【0043】

るだろうが、この状態において記録されることはできない。この場合、ロック・スレッドは、故障レコードの状態をアップデートするだろう。通常、この状態遷移は、故障スレッドのコンテキストで生じるだろう。

【0045】端末状態112「REPLICA」は、UFS故障がすでに故障したファイルシステムに生じたことを意味する。ロック・スレッドは、ufsvfsの処理が必要であるので、待ち行列状態110からこの状態にレコードを動かすことに対して責任がある。一度記録されると、修復がすでに始められたとしてこれらのレコードは無視され、オリジナルのufs_failureと関連させられる。

【0046】状態114「TRY LOCK」は、レコードがロック・スレッドによって見つけ出されて、内部のlockfs入口点(ufs_fiolf())を起動するために必要な情報によりアップデートされたことを意味する。故障したファイルシステムのufsvfsは、この点で故障がこのファイルシステムに生じたことを示すためにアップデートされる。これは、写しを作られたUFS故障を検出するために用いられる。

【0047】状態116「UMOUNT」は、ファイルシステムが修復されるよりもアンマウントされるように記録されたことを意味する。故障レコードがこの(何か他の)端末状態を成し遂げるならば、動作はとられない。

【0048】端末状態118「NOT FIX」は、このファイルシステムが修復されないと記録されたこと、または修復が自動的に生じるのを妨げるために強制的にアンマウントされたことを意味する。

【0049】状態120「LOCKED」は、故障したファイルシステムが成功裏にロックされたことを意味する。RPCコールは、修復を開始するためにポートに対してなされる。fsckが成功裏に始まるとき、RPCコールは、復帰する。

【0050】状態122「FIXING」は、fsckが修復が進行中であることを示すスーパーブロックをアップデートしたことを意味する。

【0051】端末状態124「FIXED」は、このファイルシステムがうまく修復されて、アンロックされたことを意味する。

【0052】以下は、上に記述したいくつかの状態の間の状態遷移100のための必要な条件である：

イベント102から端末状態104「PANIC」へ：非UFSパニックがすでに進行；ファイルシステムが定義可能でない(vp、ufsvfsp、vfspがNullである)；パニック回復の特性はこのファイルシステムの上で不可能だった；このファイルシステムに含まれる活動中のスワップファイルがある；そして、このファイルシステムに含まれる活動中のカーネル・アカウントファイルがある。

【0053】状態106「UNDEF」から端末状態104「PANIC」へ：ファイルシステムの制御構造(ufsvfsp、vfsp、bpがstruct fsを含んでいる)が不当になった。

【0054】状態108「INIT」から状態110「QUEUE」へ：このstruct ufs_failureは、ロック・スレッドの作業待ち行列の上に置かれた。

【0055】状態110「QUEUE」から端末状態「REPLICA」へ

CA」へ：これはこのファイルシステムと関連させられたもう一つの活動中のUFS故障である。

【0056】状態110「QUEUE」から状態114「TRY LOCK」へ：故障が認識されたので、ファイルシステム制御構造は変化しなかった。

【0057】状態110「QUEUE」から端末状態104「PANIC」へ：UFS故障の数上のシステム毎の(per-system)制限が上回った；そして、UFS故障の間での最少時間のシステム毎の(per-system)制限が上回った。

【0058】状態110「QUEUE」/114「TRY LOCK」/120「LOCKED」と122「FIXING」から端末状態118「NOT FIX」へ：このファイルシステムは、いくつかの他のスレッドによってアンマウントされた。

【0059】状態114「TRY LOCK」から端末状態104「PANIC」へ：ファイルシステムの誤りロックをためす間にタイムアウトした；lockfsはこのファイルシステムの上でエラーEDEADLKをすることに失敗した；ブート以来このファイルシステム上にあまりに多くの故障があった；そして、この故障がこのファイルシステム上の最後の故障からあまりにすぐに生じた。

【0060】状態114「TRY LOCK」から状態116「UMOUNT」へ：このファイルシステムは故障でアンマウントされるために記録される。

【0061】状態116「UMOUNT」から端末状態「NOT FIX」へ：ファイルシステムのアンマウントが成功した。

【0062】状態114「TRY LOCK」から、状態120「LOCKED」へ：ファイルシステムの誤りロックは成功した；そして、ファイルシステムの誤りロックが失敗した、これによりそれがすでに誤りロック中であることを示す。

【0063】状態120「LOCKED」から状態104「PANIC」へ：fsckが開始しない又はできない。

【0064】状態120「LOCKED」から状態122「FIXING」へ：ファイルシステムのスーパーブロックのfs_cleanフィールドが、fsckによりFSFIXへリセットされた。

【0065】状態122「FIXING」から状態120「LOCKED」へ：ファイルシステムのスーパーブロックのfs_cleanフィールドは、fsckによりFSBADへリセットされた；そして、修復デーモンがfsckの実行中にエラーを検出した。

【0066】状態122「FIXING」から端末状態124「FIXED」へ：ファイルシステム上の誤りロックはアンロックされた(これは、fs_clean == FSCLEANを必要とする)。

【0067】本発明のシステムと方法を利用することを本明細書において開示したが、独立のスレッドの平均故障時間(MTTF)が次の故障までの時間によって増加されるので全システムの可用性が増加する。さらに、そ

これらの従属するスレッドに対するMTTFと平均修復時間(MTTR)は、任意のシステムオーバーヘッド(パニック・ダンプ、ハードウェアリセット、リブートまたは他のノードへのフェイル・オーバー(failover)のような)に対してより、むしろちょうど修復に必要な時間にまで減らされる。

【0068】本発明の原則を特定のコンピューター・オペレーティング・システムに関連して説明したが、これは単なる例示であって本発明の範囲を制限するべきものではない。

【図面の簡単な説明】

【図1】 本発明の動作環境の一部を形成している汎用計算機を代表する簡略図面である；

【図2】 図1のコンピューターの一つ以上のものから作られたコンピュータ・システム上で、実行されるソフトウェア・アプリケーションの基礎をなしており、特に「ユーザ」、「カーネル」及び「UFS」層を、いろいろなオペレーション(例えばOp1-Op3)が動作することができる後者二つの間のVFSインタフェースとともに説明している妥当なユニックス・オペレーティング・システム層の単純化された概念上の図解である；

【図3】 内部のデータ構造が正常動作の間、UFS故障の前に変化する方法的単純化された概念上の表示である；

【図4】 認識されていて、他のUFSスレッドのVFS層でのブロッキングに終わっているUFS故障の概念上の表示である；

【図5】 システムとファイルシステム・パニック回復の方法の単純化された概念上の表示であり、ここでシステムは、アンマウントまたはロックされた条件に留めら

れることができ、そして、ロック・スレッドは、ファイルシステムのチェッカーを実行するために修復を開始するように進む；

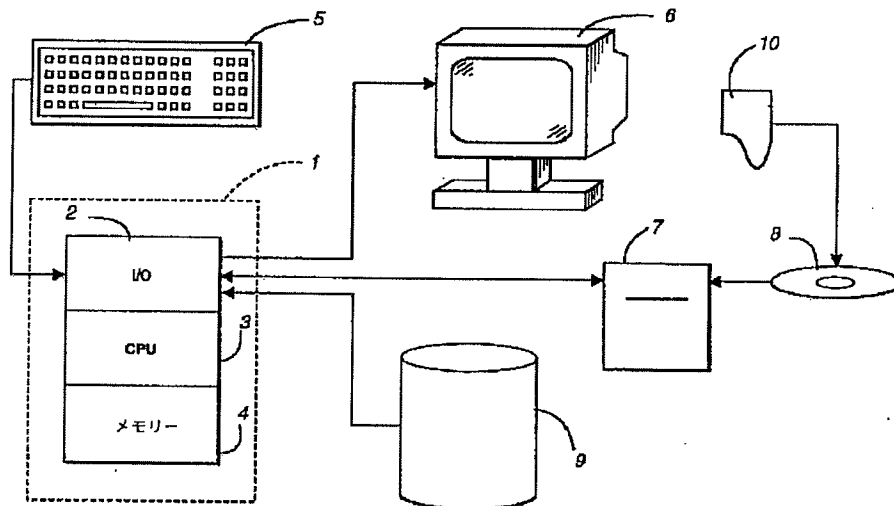
【図6】 以前の図において例を示した、層の単純化された概念上の表示であり、ここでファイルシステムのチェック「fsck」が修復を完了し、他のスレッドのオペレーションの続行が許される；

【図7】 本明細書において開示したファイルシステム・パニック回復のシステム及び方法の特定のインプリメンテーションに従うUFS故障のイベントにおける複数の状態の間でできる遷移を図示している典型的な状態図である。

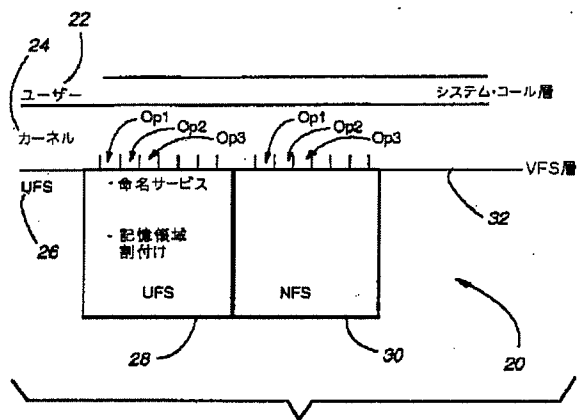
【符号の説明】

- 1：プロセッサ
- 2：入出力セクション
- 3：中央処理装置
- 4：メモリー・セクション
- 5：キーボード
- 6：表示装置
- 7：CD-ROM装置
- 8：CD-ROM媒体
- 9：磁気ディスク装置
- 10：プログラム
- 20：ユニックスOSシステム層
- 36：誤りロック
- 38、54：UFSスレッド
- 40：記憶装置
- 42：故障スレッド
- 80：他のスレッド

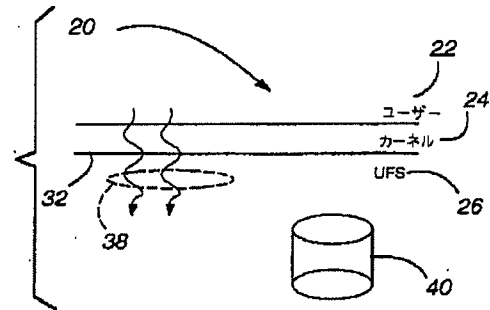
【図1】



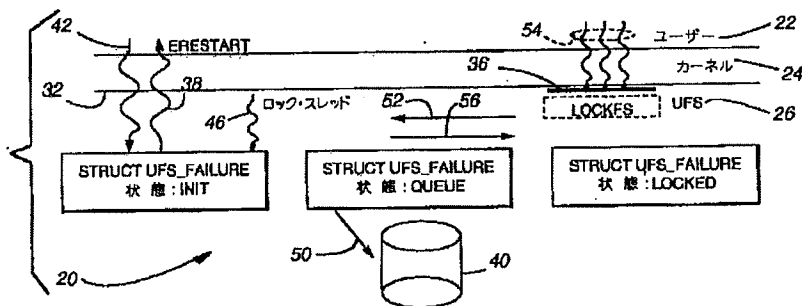
【図2】



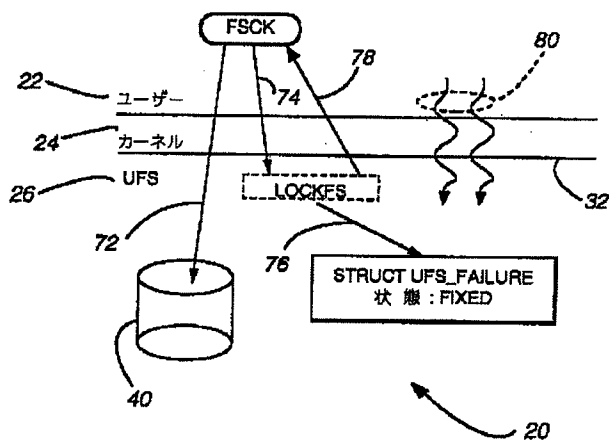
【図3】



【図4】



【図6】



[illegible]

The flowchart illustrates the state transitions of a file system during a recovery process. The states and transitions are as follows:

- Initial State:** The process starts at the top with the label "<< UFS 故障 >>" (102).
- UNDEF (106):** The first state reached after the initial fault.
- INIT (108):** The state reached after UNDEF.
- QUEUE (110):** The state reached after INIT. It is connected to PANIC (104) by a dashed line, with the label "失敗スレッド ロック・スレッド" (Failed thread lock/thread) next to it.
- TRY LOCK (114):** Reached from QUEUE. It has a self-loop labeled "116" and a transition to UNMOUNT (112).
- LOCKED (120):** Reached from TRY LOCK. It has a self-loop labeled "122" and a transition to FIXING (124).
- FIXING (124):** Reached from LOCKED. It has a self-loop labeled "120" and a transition to FIXED (124).
- FIXED (124):** The final state reached from FIXING.
- PANIC (104):** Reached from QUEUE. It has a self-loop labeled "104" and a transition back to UNDEF (106).
- NOT FIXED (118):** Reached from UNMOUNT (112), TRY LOCK (114), LOCKED (120), and FIXING (124). It has a self-loop labeled "118" and a transition back to QUEUE (110).
- REPLICA (112):** Reached from UNMOUNT (112). It has a transition back to QUEUE (110).

Additional labels include "<<RPCがリペア・プロセスをトリガする>>" (100) pointing to the LOCKED state, and "100" at the bottom left.